一元半正定多项式降次平方和表示及其L-算法*

黄勇† [1,2]　曾振柄 [3]　杨路 [4]　饶永生 [1,2]

[1] 广州大学计算科技研究院, 广州 510006
[2] 广东省数学教育软件工程技术研究中心, 广州 510006
[3] 上海大学理学院数学系, 上海 200444
[4] 中国科学院成都计算机应用研究所, 成都 610041

摘要　本文给出一元半正定多项式一种降次平方和表示方法, 并给出从已知半正定多项式得到其降次平方和表示的算法. 在第四节, 我们把这种"降次平方和"表示思路和算法用到多元多项式上也取得成功.

关键词　半正定多项式, 平方和表示, SOS, 构造性, 希尔伯特第17问题

1　引言

除非特别指明, 本文讨论的多项式和矩阵都假定是实数域$\mathbb{R}$上的.

设$p(x)$为一元实系数多项式. 如果$p(x) \geq 0, \forall x \in \mathbb{R}$, 则称$p(x)$ 为半正定多项式. 如果$p(x) > 0, \forall x \in \mathbb{R}$, 则称$p(x)$ 为正定多项式. 将半正定的一元多项式表示成一些多项式的平方和（SOS, Sum Of Squares of Polynomials）, 是Hilbert 第17问题的一部分. 容易证明（参见[1]）$p(x)$ 半正定的充分必要条件是存在实矩阵H使得$p(x) = X(HH^{\mathrm{T}})X^{\mathrm{T}}$, 其中$X = (x^n, x^{n-1}, \cdots, x, 1)$. 而$B = HH^{\mathrm{T}}$被称作$p(x)$的Gram矩阵. [1]中通过计算Gram矩阵$B$给出了$p(x)$的SOS的一种算法.

在本文中我们证明H可以取成下三角矩阵（甚至在很多情形可以取"空心的下三角矩阵"）, 并通过直接计算H（而不是B）给出一种SOS算法. 在该算法里, 只需要解一个自然三角化了的线性方程组, 所涉及的运算均为有理运算, 只涉及很初等的理论知识, 大学低年级的学生甚至中学生都能理解. 与[1] 比较, 本文算法不需要计算矩阵的特征多项式, 不需要做实对称矩阵对角化, 也不需要使用关于多项式根的Descarte符号法则. 在做半单代数集计算时, 我们的方法只需计算一个多项式（有理函数）, [1]中需要处理多个多项式相关的问题.

关于一元半正定多项式的其它SOS表示, 可以参阅[3]. 关于多元（包括一元）半正定多项式的SOS 表示的较新进展的综述性论文, 可以参阅[5, 6, 10].

*ACKNOWLEDGMENTS: 国家自然科学基金(批准号: 11701118)及国家重点研发计划（批准号: 2018YFB1005104）资助项目

†电子邮件: yongh@gzhu.edu.cn

2 理论与算法

定理1: （存在性定理）一元多项式$p(x)=\sum_{k=0}^{2n}c_kx^k$半正定的充分必要条件是它能表示成

$$p(x)=\sum_{i=0}^{n}\left(\sum_{j=0}^{i}a_{i,j}x^j\right)^2. \tag{1}$$

其中$a_{i,i}\geq 0, 1\leq i\leq n$.

证明: 充分性显然成立, 因此只证明必要性.

不妨假定$p(x)$首项系数为1, 由半正定性, $p(x)$有n对共轭复根$\alpha_k\pm\beta_k\,\mathrm{i}$, $k=1,2,\cdots,n$, 其中α_k,β_k均为实数（半正定多项式的实根一定是偶数重的, 它的任意一个2重实根我们可以看成是这里$\beta_k=0$时的特殊情况）, i 是虚数单位, 满足$\mathrm{i}^2=-1$, 从而

$$p(x)=\prod_{k=0}^{n}\left[(x-\alpha_k)^2+\beta_k^2\right]=s_n^2(x)+q(x). \tag{2}$$

其中$s_n(x)=\prod_{k=0}^{n}(x-\alpha_k)$是一个次数为$n$的多项式, 而

$$q(x)=\sum_{k=1}^{n}\left(\frac{\beta_k}{x-\alpha_k}s_n(x)\right)^2+\sum_{1\leq k,l\leq n,k\neq l}\left(\frac{\beta_k\beta_l}{(x-\alpha_k)(x-\alpha_l)}s_n(x)\right)^2+\cdots+\left(\prod_{k=0}^{n}\beta_k\right)^2. \tag{3}$$

当$p(x)$的根全为实数时, $q(x)$恒等于0（β_k全为0）, 否则$q(x)$是次数为$2n-2$ 的半正定多项式（至少一个β_k不为0）. 当$p(x)$正定时（所有的$\beta_k\neq 0$）, $q(x)$ 也是正定的.

由上述已证得的事实, 利用数学归纳法, 可得定理的必要性. □

用矩阵的形式来看定理1会更直观.

推论2: （存在性定理, 矩阵形式）一元多项式$p(x)=\sum_{k=0}^{2n}c_kx^k$半正定的充分必要条件是它能表示成

$$p(x)=X(HH^{\mathrm{T}})X^{\mathrm{T}}. \tag{4}$$

其中$X=(x^n,x^{n-1},\cdots,x,1)$, 而$H$是下三角矩阵, 它具有形状

$$H=\begin{pmatrix} a_{n,n} & & & \\ a_{n,n-1} & a_{n-1,n-1} & & \\ \vdots & \vdots & \ddots & \\ a_{n,0} & a_{n-1,0} & \cdots & a_{0,0} \end{pmatrix} \tag{5}$$

其中$a_{i,i}\geq 0, 1\leq i\leq n$.

根据定理1的证明过程, 我们可以得到H的以下性质.

推论3: 如果$p(x)$是正定的, 则所有的$a_{i,i}>0$; 如果$p(x)$是半正定的, 则存在一个i_0, 当$i>i_0$ 时$a_{i,i}>0$, 而$i\leq i_0, j\leq i$ 时$a_{i,j}=0$.

推论3表明, 矩阵H的主对角线上的元素全部都是非负实数. 如果这些对角线上元素不全大于0, 一定存在一个最左的0, 在这一列及其右边的所有列的元素都是0, 而在这一列的

左边的主对角线上的元素全部大于0. 这说明平方和表示中, 可以“断尾巴”, 不能“掉脑袋”. 同时, 推论3还表明, 定理1 所述的平方和的次数是严格递降的, 即右端的平方项的次数依次是相邻的偶数$2n, 2n-2, \cdots, 2n_0$, 其中n_0是满足$0 \le n_0 < n$的整数. 因此(1) 不会出现$2x^2 = x^2 + x^2$ （两项的次数相等）, 也不会出现如$x^6 + x^2$ 这样的表示（缺4次项）, 根据我们的算法后者的SOS可以表示为$x^6 + x^2 = \left(x^3 - \frac{1}{2}x\right)^2 + x^4 + \frac{3}{4}x^2$.

为讨论方便, 我们将(5)中的待定参数分成三类:

$$\begin{aligned} L :=& (a_{n,n}, a_{n,n-1}, \cdots, a_{n,0}, a_{n-1,0}, \cdots, a_{1,0}, a_{0,0}), \\ D :=& (a_{i,i} | 0 < i < n), \\ V :=& (a_{i,j} | 1 < i < n, 0 < j < i). \end{aligned} \tag{6}$$

第一类L是位于第1列和第$n+1$行的那$2n+1$个未知量, 是我们待求解的参数; 第二类D是位于对角线上的参数(去掉两端), 这些参数是半自由的; 第三类V是余下部分（位于三角形内部）, 这些参数是自由的. 在下文中, $D > 0$是指D中每个变元取值大于0 , 类似地$V = 0$ 是指V中每个变元取值等于0.

Algorithm 1 正定多项式SOS算法

输入: 正定多项式$p(x) = \sum_{k=0}^{2n} c_k x^k$.

输出: 形如公式(1)的SOS.

1: 比较(1)两端x各次幂的系数, 得出方程组

$$\left\{ \begin{aligned} c_{2n} &= a_{n,n}^2 \\ c_{2n-1} &= 2a_{n,n}a_{n,n-1} \\ &\vdots \\ c_n &= 2a_{n,n}a_{n,0} + \cdots \\ c_{n-1} &= 2a_{n-1,n-1}a_{n-1,0} + \cdots \\ &\vdots \\ c_1 &= 2a_{1,1}a_{1,0} + \cdots \\ c_0 &= a_{0,0}^2 + \cdots. \end{aligned} \right. \tag{7}$$

2: 在条件$D > 0$下, 按顺序从(7)的第k个方程解出L中的第k 个未知量. 最后一个方程不必解$a_{0,0}$, 只需要解出$A002 := a_{0,0}^2$.

3: 取非空半代数集$S := \{A002(D,V) > 0, D > 0\}$ 中一个点D_0, V_0, 并计算$L_0 = L(D_0, V_0)$. 将L_0, V_0, D_0的值回代到(1).

4: **return** 形如公式(1)的SOS.

Algorithm 1中第3步里的S的非空性由推论3得到保证. 在Algorithm 1中, 一个重要的特征是待求解的变元集合L中的变元位于矩阵H的第一列和最后一行, 正好呈现"L"形, 我们不妨将其称为L-算法.

方程组(7)有如下特点: 1）已经自然三角化, 即L中的第k个未知量在(7)的第k个方程才首次出现; 2）除开首尾两个方程外, L中的第k个未知量在第k个方程中是线性的, 它在该方程的"$\cdots$"部分不会出现. 中间的$2n-1$个是线性方程容易求解（注意条件$a_{k,k}>0, 1\le k\le n$）. 第一个方程关于$a_{n,n}$是2 次的只是计算算术平方根, 但通常我们可以将$p(x)$ 的首项系数规范化为1, 从而$a_{n,n}=1$, 这样可以规避掉这个开方运算, 并且使前$n+1$ 个方程求解也变得更简单. 最后一个方程, 看上去必须作开平方运算, 其实不然, 只需要解出$A002 := a_{0,0}^2$（而不是$a_{0,0}$, 回代时也直接回代$A002$, 这样完全避开开方运算）.

推论4: 首1多项式$p(x)=\sum_{k=0}^{2n} c_k x^k$正定的充要条件是半代数集

$$\{A002(D,V)>0, D>0\}\neq\emptyset.$$

次数不超过2的正定多项式, 表示成降次平方和时, 因为不含有待定的参数, 所以只有唯一的降次平方和表示. 利用连续函数的保号性, 我们还有

推论5: 次数大于2的正定多项式的降次平方和表示有无穷多个.

半正定而非正定的情形按以下方法处理.

Algorithm 2 半正定多项式SOS算法

输入: 半正定多项式$p(x)=\sum_{k=0}^{2n} c_k x^k$.

输出: $p(x)$的SOS.

1: 作无平方因子分解$p(x)=(f(x))^2 g(x)$, 其中$g(x)$无平方因子因而是正定的.
2: 将$g(x)$按Algorithm 1 写出SOS.
3: 把$p(x)$写成SOS.
4: **return** 上一步中得出的SOS.

推论6: 次数大于2的半正定多项式只有唯一的降次平方和表示的充要条件是它的根全部为实数.

在Algorithm 1中, $A002 := a_{0,0}^2$是D,V的变元的有理函数$A002(D,V)$. L中的各参数也是D,V 的有理函数. 由连续函数的保号性和有理数的稠密性, 我们可以取那些使得$A002>0$的待定参数为有理数, 这样在$p(x)$ 是首1 多项式的情况下, 我们可以保证求出来的$a_{i,j}, j\le i$都是有理数, 唯一的例外是$a_{0,0}$ 未必是有理数, 但我们可以保证$A002$是有理数.

推论7: 设$p(x)$是首1的有理系数正定多项式, 则存在有理系数的降次平方和表示

$$\sum_{i=0}^{n}\left(\sum_{j=0}^{i} a_{i,j}x^j\right)^2.$$

其中除$a_{0,0}$外（但$a_{0,0}^2$是有理数），其余所有系数 $a_{i,j}$ 均为有理数.

推论8: 设$p(x)$是有理系数半正定多项式, 则存在有理系数的加权降次平方和表示

$$\sum_{i=0}^{n} d_i \left(\sum_{j=0}^{i} a_{i,j} x^j \right)^2$$

所有系数 $d_i(\geq 0), a_{i,j}$ 均为有理数.

3 计算实例

例: 对于$x \in \mathbb{R}$, 求证$x^6 - x^5 - 2x^4 + x^3 + x^2 + 1 > 0$.

解: 这是一个经许多数学爱好者用来讨论正定性判定的一个例子[13], 以下的SOS表示便是正定性的一个“明证”.

$$\begin{aligned} & x^6 - x^5 - 2x^4 + x^3 + x^2 + 1 \\ & = \left(x^3 - \frac{1}{2}x^2 - \frac{5}{4}x - \frac{1}{8} \right)^2 + \left(\frac{1}{2}x^2 - \frac{15}{16} \right)^2 + \left(\frac{1}{2}x - \frac{5}{16} \right)^2 + \frac{1}{128}. \end{aligned} \tag{8}$$

□

例: 本例来自文献[3]. 写出$x^4 + 2\,x^3 - 18\,x^2 - 12\,x + 117$ 的SOS表示.

解: 表示(9)来自文献[3].

$$\begin{aligned} & x^4 + 2\,x^3 - 18\,x^2 - 12\,x + 117 \\ & = \left(x^2 + x - \frac{217046}{21315} \right)^2 + \frac{29107}{21315} \left(x + \frac{89156}{29107} \right)^2 + \frac{6597622612523}{13224160752075}. \end{aligned} \tag{9}$$

用本文算法得到一个表示如(10)

$$x^4 + 2\,x^3 - 18\,x^2 - 12\,x + 117 = \left(x^2 + x - 10 \right)^2 + (x + 4)^2 + 1. \tag{10}$$

□

例: 本例来自文献[16], 写出$x^6 - 2\,x^5 + 4\,x^4 - 6\,x^3 + 6\,x^2 - 4\,x + 2$ 的SOS表示.

解: 本文算法得到一个表示如(11)

$$\begin{aligned} & x^6 - 2\,x^5 + 4\,x^4 - 6\,x^3 + 6\,x^2 - 4\,x + 2 \\ & = \left(x^3 - x^2 + x - 1 \right)^2 + \left(x^2 - x + \frac{1}{2} \right)^2 + \left(x - \frac{1}{2} \right)^2 + \frac{1}{2}. \end{aligned} \tag{11}$$

□

Algorithm 1 给出的正定多项式平方和的缺点是有时把一个“简单”的多项式会写成“复杂”的形式.

例: 写出$x^6 + 1$的SOS表示.

解: 根据算法可以得到一个SOS表示

$$x^6+1=\left(x^3-\frac{1}{8}x\right)^2+\left(\frac{1}{2}x^2-\frac{17}{64}\right)^2+\frac{1}{4}x^2+\frac{3807}{4096}. \tag{12}$$

□

例: (半正定多项式) 写出$x^6-6\,x^5+14\,x^4-18\,x^3+17\,x^2-12\,x+4$ 的SOS表示.

解: 事实上

$$x^6-6\,x^5+14\,x^4-18\,x^3+17\,x^2-12\,x+4=\left(x^2+1\right)(x-1)^2\,(x-2)^2.$$

是半正定的, 按照Algorithm 2 我们得到SOS表示如下

$$x^6-6x^5+14x^4-18x^3+17x^2-12x+4=\left(x^3-3x^2+2x\right)^2+\left(x^2-3x+2\right)^2. \tag{13}$$

□

例: (稀疏和非稀疏表示) 给出$x^{10}-x+1$ 的SOS表示.

解: 这里给出两种表示, 表示(14)是预先取定$V_0=0$给出的稀疏表示(空心三角形矩阵表示, V中的10个变元都预先取值为0, 节省了很多计算量), 而(15)则是未预先取定自由参数的值的非稀疏表示.

$$\begin{aligned}&x^{10}-x+1\\&=\left(x^5-\frac{1}{2}x^3-\frac{1}{4}x\right)^2+\left(x^4-\frac{5}{8}\right)^2+\frac{1}{4}x^6+\left(x^2-\frac{17}{32}\right)^2+\left(x-\frac{1}{2}\right)^2+\frac{79}{1024}.\end{aligned} \tag{14}$$

$$\begin{aligned}&x^{10}-x+1\\&=\left(x^5-\frac{1}{2}\,x^3+x^2-\frac{3}{4}\,x-\frac{1}{4}\right)^2+\left(x^4-x^3+\frac{1}{2}x-\frac{5}{8}\right)^2\\&\quad+\left(\frac{1}{2}\,x^3+\frac{1}{2}\,x^2-\frac{1}{2}\right)^2+\left(\frac{1}{2}\,x^2+\frac{1}{2}x-\frac{5}{16}\right)^2+\left(\frac{1}{2}x-\frac{7}{16}\right)^2+\frac{1}{128}.\end{aligned} \tag{15}$$

□

但这种类型的“稀疏”表示并非总是存在.

例: $x^6-2\,x^5+5\,x^2-4\,x+1$具有SOS表示

$$\begin{aligned}&x^6-2\,x^5+5\,x^2-4\,x+1\\&=\left(x^3-x^2-x+\frac{1}{2}\right)^2+\left(x^2-\frac{3}{2}\,x+\frac{1}{4}\right)^2+\left(\frac{3}{2}\,x-\frac{3}{4}\right)^2+\frac{1}{8}.\end{aligned} \tag{16}$$

但是它不存在“空心三角矩阵”形式的SOS表示.

解: 事实上, 假定它有空心三角矩阵形式的表示, 则按Algorithm 1 算出来的

$$A002 = (4\,s^{10} + \left(t^2+16\right)s^8 + \left(28\,t^2+88\right)s^6 + \left(8\,t^4+38\,t^2+144\right)s^4 \\ + \left(48\,t^4-228\,t^2+324\right)s^2 + (16\,t^6-120\,t^4+225\,t^2))/(-64s^2t^2).$$

其中$s = a_{2,2}, t = a_{1,1}$. 由于$48\,t^4-228\,t^2+324>0, 16\,t^6-120\,t^4+225\,t^2 = t^2\left(4\,t^2-15\right)^2 \geq 0$, 所以$A002<0$在$s>0, t>0$ 时恒成立. □

对于次数比较高的多项式的SOS表示, 主要在计算半代数集中的点时耗费很多时间, 某些情况下做一些特殊处理, 可以减少计算量.

例: 写出$p(x) = 2x^{16} - 4x^{15} - 2x^{14} + 4x^{13} + 2x^{12} - x^5 + 7x^4 - 9x^3 - 7x^2 + 9x + 6$的SOS表示.

解: 通过求$p(x)$导数的实根, 我们可以求出它的最小值等于1, 由此容易求得到分解$p(x)-1 = (x^2-x+1)^2(2x^{12}-x+5)$, 其中$2x^{12}-x+5$是一个无重因子的正定多项式, 先利用前面的算法求出$2x^{12}-x+5$ 的平方和表示, 然后可以得出$p(x)$的SOS表示如下.

$$\begin{aligned}
&2x^{16} - 4x^{15} - 2x^{14} + 4x^{13} + 2x^{12} - x^5 + 7x^4 - 9x^3 - 7x^2 + 9x + 6 \\
&\quad = 2\left(x^8 - x^7 - \frac{3}{2}x^6 + \frac{1}{2}x^5 + \frac{1}{4}x^4 + \frac{1}{4}x^3 + \frac{1}{2}x^2 - \frac{1}{4}x - \frac{1}{4}\right)^2 \\
&\quad + 2\left(x^7 - x^6 - x^5 + \frac{1}{2}x^4 - \frac{3}{2}x^3 + \frac{1}{4}x^2 + \frac{5}{4}x + \frac{1}{4}\right)^2 \\
&\quad + 2\left(\frac{1}{2}x^6 - \frac{3}{2}x^5 + \frac{1}{2}x^4 + \frac{3}{2}x^3 - \frac{9}{16}x^2 - \frac{7}{16}x + \frac{1}{16}\right)^2 \\
&\quad + 2\left(\frac{1}{2}x^5 - \frac{1}{2}x^4 - \frac{1}{2}x^3 - \frac{1}{8}x^2 + \frac{1}{8}x + \frac{1}{8}\right)^2 \\
&\quad + 2\left(x^4 - \frac{1}{2}x^3 - \frac{41}{16}x^2 + \frac{9}{16}x + \frac{17}{16}\right)^2 \\
&\quad + 2\left(x^3 - \frac{15}{16}x^2 - \frac{17}{16}x - \frac{1}{16}\right)^2 \\
&\quad + \frac{313}{128}\left(x^2 - x - 1\right)^2 \\
&\quad + 1.
\end{aligned} \tag{17}$$

对于一个正定的16次多项式$p(x)$, 按照Algorithm 1直接计算, 会出现120个待定参数, 假定每个参数取2 个不同的值构成一个网格进行搜索, 那么网格的点数会达到1.3×10^{36}, 要搜索完这样一个网格, 是一个非常艰难的任务. 正定多项式一定是可以取到最小值的, 如果我们有办法求出$p(x)$的最小值b （必定大于0）, 则$p(x)-b$ 将是一个半正定多项式, 并且$p(x)-b$将有偶数重的实根. 设$p(x)-b = f(x)(g(x))^2$, 其中f是一个无重因子的正定多项式, 它的次数至少比$p(x)$ 低2次, 此时可以先做出f的平方和表示, 然后再做出$p(x)$ 的平方和表示, 往往可以大大降低计算难度. 这个例子里, 16次正定多项式, 当我们知道它的

最小值后, 分离出来两对2重实根, 这样次数降低4, 变成一个实质上的12次多项式问题, 对应得待定参数减少到66个, 几乎是原来的一半. 而同样的每个变元取2值的网格格点数只有7.4×10^{19} 个. □

不过正定多项式计算最小值是很难的, 这种方法并不是对所有正定多项式都有效.

4 多元多项式的SOS

对于多元多项式的SOS问题, 我们也可以转换成一元多项式进行处理.

Algorithm 3 多元半正定多项式SOS算法

输入: 多元半正定多项式$p(x,y,\cdots,z)$.

输出: $p(x,y,\cdots,z)$的SOS表示, 或"不能表示成多项式的平方和".

1: 按如下方法将$p(x,y,\cdots,z)$转换成一元多项式$q(t)$: 取$n_x:=1$, 计算$p(t^{n_x},y,\cdots,z)$中t的次数记为$n_y-1,\cdots$, 计算$p(t^{n_x},t^{n_y},\cdots,z)$ 中t的次数记为n_z-1. 令

$$q(t)=p(t^{n_x},t^{n_y},\cdots,t^{n_z}). \tag{18}$$

2: 挑选会在$p(x,y,\cdots,z)$平方和中出现的单项式列表, 并计算将在$q(t)$的平方和表示中出现的t的各次幂(按降幂排列)$U=(t^{k_n},t^{k_{n-1}},\cdots,t^{k_1})$.

3: 假设$q(t)$具有（降次）SOS形式:

$$q(t)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}a_{i,j}t^{k_j}\right)^2. \tag{19}$$

4: 比较$q(t)$的两个表达式中t的各次幂得出方程组. 如果该方程组无（实数）解, 则返回"不能表示成多项式的平方和". 如果有解, 则将解代回上式得到$q(t)$ 的SOS 表示如(19).

5: 在(19)的两端顺次施行如下运算

$$\operatorname{rem}(\cdot,t^{n_z}-z,t),\ \cdots,\ \operatorname{rem}(\cdot,t^{n_y}-y,t),\operatorname{rem}(\cdot,t^{n_x}-x,t). \tag{20}$$

反过来得出$p(x,y,\cdots,z)$的SOS表示.

6: **return** 上一步中得出的SOS.

注: 对于在(19)中SOS仍然取下三角矩阵形式, 其存在性可以通过半正定矩阵的Cholesky分解来实现[15]. Cholesky分解是说, 对于一个半正定矩阵B一定存在下三角矩阵H使得$B=HH^{\mathrm{T}}$, 并且H的对角线上的元素均非负. 如果B 正定, 则H的对角线上的元素均大于0. 实际上，定理1、推论2和推论3也可以利用Cholesky分解得到.

例: （来自[1]）将3元多项式$f(x,y,z)=x^6+4\,x^3y^2z+y^6+2\,y^4z^2+y^2z^4+4\,z^6$ 给出平方和表示.

解: 注意x, y, z的最高次数分别为6,令

$$g(t) = f(t, t^7, t^{43}) = 4\,t^{258} + t^{186} + 2\,t^{114} + 4\,t^{60} + t^{42} + t^6.$$

注意到$f(x, y, z)$的平方和表示中只能出现如下的项: $[x^3, y^3, y^2z, yz^2, z^3]$. 从而$g(t)$的平方和表示中只能出现以下项$[t^3, t^{21}, t^{57}, t^{93}, t^{129}]$. 设$g(t)$的SOS有待定形式

$$\begin{aligned} g(t) = & \left(a_4t^{129} + a_3t^{93} + a_2t^{57} + a_1t^{21} + a_0t^3\right)^2 \\ & + \left(b_3t^{93} + b_2t^{57} + b_1t^{21} + b_0t^3\right)^2 \\ & + \left(c_2t^{57} + c_1t^{21} + c_0t^3\right)^2 \\ & + \left(d_1t^{21} + d_0t^3\right)^2 \\ & + (e_0t^3)^2. \end{aligned}$$

比较$g(t)$的两种表示的关于t的各次幂的系数, 得到一个关于这些待定系数的方程组, 求得方程组的一个解$a_4 = 2, a_3 = 0, a_2 = 0, a_1 = 0, a_0 = 0, b_3 = 1, b_2 = 0, b_1 = -1, b_0 = 0, c_2 = 2, c_1 = 0, c_0 = 1, d_1 = 0, d_0 = 0, e_0 = 0$. 因此得到$g(t)$一个SOS表示

$$4\,t^{258} + t^{186} + 2\,t^{114} + 4\,t^{60} + t^{42} + t^6 = (2\,t^{129})^2 + (t^{93} - t^{21})^2 + (2\,t^{57} + t^3)^2. \quad (21)$$

在上式两端**顺次**施行如下求余运算

$$\mathrm{rem}(\,\cdot\,, t^{43} - z, t), \mathrm{rem}(\,\cdot\,, t^7 - y, t), \mathrm{rem}(\,\cdot\,, t - x, t). \quad (22)$$

则得到原多元多项式$f(x, y, z)$如下SOS表示

$$x^6 + 4\,x^3y^2z + y^6 + 2\,y^4z^2 + y^2z^4 + 4\,z^6 = (2\,z^3)^2 + (z^2y - y^3)^2 + (x^3 + 2\,y^2z)^2. \quad (23)$$

□

下一个例子也是来自[1], 原文中对本问题的结果是否定的, 即判断不能表示成SOS. 有关不能表示成多项式的平方和的问题可以参阅[2].

例: (来自[1])给出$f(x, y, z) = x^4 + x^3z + 2\,x^2y^2 + z^4$ 的SOS表示.

解: 因x的最高次数为4, x, y的混合最高次数项为x^2y^2, z的最高次数为4. 令$g(t) := f(t, t^5, t^{13}) = t^{52} + t^{16} + 2\,t^{12} + t^4$. 在$f$的SOS 表示中会出现的项为$[x^2, yx, xz, z^2]$. 在$g(t)$的SOS表示中会出现的项为$[t^2, t^6, t^{14}, t^{26}]$. 假设$g(t)$具有以下形式

$$\begin{aligned} g(t) = & \left(a_3t^{26} + a_2t^{14} + a_1t^6 + a_0t^2\right)^2 \\ & + \left(b_2t^{14} + b_1t^6 + b_0t^2\right)^2 \\ & + \left(c_1t^6 + c_0t^2\right)^2 \\ & + (d_0t^2)^2. \end{aligned}$$

比较$g(t)$两个表达式的系数得到一个方程组，求得方程组的一个解$a_3 = 1, a_2 = 0, a_1 = 0, a_0 = -1/2, b_2 = 1, b_1 = 0, b_0 = 1/2, c_1 = \sqrt{2}, c_0 = 0, d_0 = \sqrt{2}/2$. 从而我们得到SOS表示

$$t^{52} + t^{16} + 2\,t^{12} + t^4 = \left(t^{26} - \frac{1}{2}\,t^2\right)^2 + \left(t^{14} + \frac{1}{2}\,t^2\right)^2 + 2\,t^{12} + \frac{1}{2}\,t^4. \tag{24}$$

在上式两端**顺次**施行如下求余运算

$$\mathrm{rem}(\,\cdot\,, t^{13} - z, t),\ \mathrm{rem}(\,\cdot\,, t^5 - y, t),\ \mathrm{rem}(\,\cdot\,, t - x, t). \tag{25}$$

则得到原多元多项式$f(x, y, z)$如下SOS表示

$$x^4 + x^3 z + 2\,x^2 y^2 + z^4 = \left(z^2 - \frac{1}{2}\,x^2\right)^2 + \left(xz + \frac{1}{2}\,x^2\right)^2 + 2\,x^2 y^2 + \frac{1}{2}\,x^4. \tag{26}$$

□

例：（来自[4]）证明 $f(x, y) = x^6 + 2\,x^5 y + 5\,x^2 y^4 + 4\,xy^5 + y^6 \geq 0$.

解： 因x, y的最高次数都是6，令

$$g(t) = f(t, t^7) = t^{42} + 4\,t^{36} + 5\,t^{30} + 2\,t^{12} + t^6.$$

因为在$f(x, y)$的SOS表示中只能出现单项式$[x^3, x^2y, xy^2, y^3]$，所以在$g(t)$的SOS表示中只能出现单项式$[t^3, t^9, t^{15}, t^{21}]$. 假设

$$\begin{aligned} g(t) = & \left(a_3 t^{21} + a_2 t^{15} + a_1 t^9 + a_0 t^3\right)^2 \\ & + \left(b_2 t^{15} + b_1 t^9 + b_0 t^3\right)^2 \\ & + \left(c_1 t^9 + c_0 t^3\right)^2 \\ & + (d_0 t^3)^2 \end{aligned}$$

比较$g(t)$两个不同的表示的系数得方程组，求得方程组得一组解为$a_3 = 1, a_2 = 2, a_1 = 0, a_0 = -1/2, b_2 = 1, b_1 = 1/2, b_0 = -1/4, c_1 = 3/2, c_0 = 3/4, d_0^2 = 1/8$. 所以我们得到$g(t)$的表示

$$\begin{aligned} & t^{42} + 4\,t^{36} + 5\,t^{30} + 2\,t^{12} + t^6 \\ & = \left(t^{21} + 2\,t^{15} - \frac{1}{2}\,t^3\right)^2 + \left(t^{15} + \frac{1}{2}\,t^9 - \frac{1}{4}\,t^3\right)^2 + \left(\frac{3}{2}\,t^9 + \frac{3}{4}\,t^3\right)^2 + \frac{1}{8}t^6. \end{aligned} \tag{27}$$

在上式两端**顺次**施行如下求余运算

$$\mathrm{rem}(\,\cdot\,, t^7 - y, t),\ \mathrm{rem}(\,\cdot\,, t - x, t). \tag{28}$$

则得到原多元多项式$f(x, y)$如下SOS表示

$$\begin{aligned} & x^6 + 2\,x^5 y + 5\,x^2 y^4 + 4\,xy^5 + y^6 \\ & = \left(y^3 + 2\,xy^2 - \frac{1}{2}\,x^3\right)^2 + \left(xy^2 + \frac{1}{2}\,x^2 y - \frac{1}{4}\,x^3\right)^2 + \left(\frac{3}{2}\,x^2 y + \frac{3}{4}\,x^3\right)^2 + \frac{1}{8}\,x^6. \end{aligned} \tag{29}$$

□

例: (Motzkin的著名例子) 证明$f(x,y)=x^4y^2+x^2y^4-3\,x^2y^2+1$ 不能表示成多项式的平方和.

解: 由算术几何平均不等式知$f(x,y)\geq 0$始终成立.

$$g:=f(t,t^5)=t^{22}+t^{14}-3\,t^{12}+1$$

$$h:=[1,x,y,yx,x^2y,xy^2]$$

$$L:=[1,t,t^5,t^6,t^7,t^{11}]$$

$$g1:=\sum_{j=1}^{6}\left(\sum_{i=1}^{j}a_{j,j-i+1}L_{j-i+1}\right)^2$$

比较g和$g1$的系数, 我们容易确定出$a_{6,6}=1$, $a_{6,5}=0$, $a_{6,4}=0$, $a_{6,3}=0$, $a_{5,5}=1$, $a_{5,4}=0$, $a_{5,2}=0$, $a_{6,2}=-1/2\,{a_{4,4}}^2-a_{5,3}-3/2$, $a_{6,1}=-a_{4,3}a_{4,4}$, $a_{5,3}=0$, $a_{4,3}=0$, $a_{3,3}=0$. 此时有

$$\begin{aligned}g1-g=&(2\,a_{4,2}a_{4,4}+2\,a_{5,1})t^7+2a_{4,1}a_{4,4}\,t^6\\&+(\frac{1}{4}\,{a_{4,4}}^4+\frac{3}{2}\,{a_{4,4}}^2+\frac{9}{4}+{a_{2,2}}^2+{a_{3,2}}^2+{a_{4,2}}^2)t^2\\&+(2\,a_{2,1}a_{2,2}+2\,a_{3,1}a_{3,2}+2\,a_{4,1}a_{4,2})t\\&+{a_{1,1}}^2+{a_{2,1}}^2+{a_{3,1}}^2+{a_{4,1}}^2+{a_{5,1}}^2-1.\end{aligned}\tag{30}$$

等式(30)右端t^2的系数不可能等于0, 从而$g1-g=0$不可能成立, 这说明$x^4y^2+x^2y^4-3\,x^2y^2+1$不可能表示成多项式平方和. □

5 补充说明和讨论

在[7]中有专门的计算半代数集的软件包SDPTools可以供大家使用, 另可参考[7, 9]. 至于无平方因子的分解算法, 则可以参见[8].

参考文献

[1] Powers V , Wörmann T . An algorithm for sums of squares of real polynomials[J]. Journal of Pure & Applied Algebra, 1998, 127(1):99-104.

[2] 冯克勤. 关于不是多项式平方和的半正定型[J]. 数学学报, 1982(1).

[3] Menini L , Tornambe A . Exact Sum Of Squares decomposition of univariate polynomials[C]// IEEE Conference on Decision & Control. IEEE, 2015:1072-1077.

[4] 姚勇, 冯勇. 一类半正定多项式的平方和分解及其表达式的自动生成[J]. 计算机学报, 2006, 029(010):1862-1868.

[5] Oliveira M D . Decomposition of a Polynomial as a Sum-of-Squares of Polynomials and the S-Procedure[C]// European Control Conference Cdc-ecc 05 IEEE Conference on Decision & Control. IEEE, 2006.
[6] Powers V. http://www.mathcs.emory.edu/vicki/preprint/PsdSosSurvey.pdf Positive Polynomials and Sums of Squares: Theory and Practice. 2015.
[7] 郭峰. SDPTools: 基于Maple的高精度求解SDP软件包[J]. 系统科学与数学(11):1512-1521.
[8] Gathen J V Z , Jürgen Gerhard. Modern computer algebra (2. ed.)[M]. DBLP, 2003.
[9] Grigoriev D , Vorobjov N . Solving systems of polynomial inequalities in subexponential time[J]. Journal of Symbolic Computation, 1988, 5(1):37-64.
[10] Xia B , Yang L . Automated Inequality Proving and Discovering (Chapter 9. SOS Decomposition)[M]. 2016, 10.1142/9951:251-270.
[11] Khachiyan, L. and L. Porkolab. Computing integral points in convex semi-algebraic sets. Proceedings 38th Annual Symposium on Foundations of Computer Science (1997): 162-171.
[12] El Din M S , Zhi L . Computing Rational Points in Convex Semialgebraic Sets and Sum of Squares Decompositions[J]. Siam Journal on Optimization, 2010, 20(6):2876-2889.
[13] 郑小彬. https://mp.weixin.qq.com/s/WKYcnvKiIadxmhcaTUopUg 一个6次多项式恒正的证明, 2020.
[14] M.D. Choi, T.Y. Lam and B. Reznick, Sums of squares of real polynomials, Symp. on Pure Math., Vol. 58, American Mathematical Society (Providence, Rl, 1995) 103-126.
[15] Horn R A , Johnson C R . Matrix Analysis. Cambridge University Press[M]. New York, 1985.
[16] 冯贝叶. 把正定六次多项式表为多项式的平方和的一种算法[J]. 应用数学进展, 2020, 9(3).

最后修改时间：2020年10月10日